PIL DIAGNOSTIC ROM PRELIMINARY ERS

STEVE CHOU
EXT 2070
DEC. 01, 1980

TABLE OF CONTENTS

1.0 PURPOSE OF THE ROM

THE PURPOSE OF THIS 4K ROM IS TO PROVIDE AN EASY TO USE AND INEXPENSIVE DIAGNOSTIC TOOL FOR ALL THE PRESENT AND FUTURE PIL DEVICES. A WAY TO PROVIDE THIS TOOL IS TO MAKE ALL THE PIL COMMAND AND DATA TRANSFER CAN BE EXECUTED BY THE USER LEVEL LANGUAGE. THEREFORE ANY PIL TEST PROGRAM CAN BE WRITTEN IN USER LEVEL LANGUAGE. IT WILL BE EASY TO DEVELOP AND EASY TO MODIFY. ANY FUTURE PIL PRODUCT CAN USE THE SAME DIAGNOSTIC ROM TO DEVELOP ITS OWN TEST PROGRAM TOO.

ANOTHER INTENTION OF THIS ROM IS TO MAKE THE 41C BECOMING A PIL LOOP MONITOR. USE THE 41C AS A SCOPE TO SHOW OR BUFFER THE FRAME IN THE LOOP. WITH A PRINTER PLUG-IN, FRAMES CAN BE PRINTED TOO.

2.0 DIRECT READ/WRITE TO PIL CHIP REGISTERS

THE PIL CHIP HAS 8 SINGLE BYTE REGISTERS. EACH OF THIS REGISTER CAN BE READ OR WRITTEN THROUH THE 41C X REGISTER.

WREG : WRITE TO A GIVEN REGISTER

Y IS TAKEN AS THE REGISTER NUMBER AND X IS TAKEN AS A DECIMAL NUMBER (0-255) TO WRITE TO THE REGISTER.

RREG : READ FROM A GIVEN REGISTER

X IS TAKEN AS THE REGISTER NUMBER TO READ FROM. THE CONTENT OF THE REGISTER WILL BE CONVERTED INTO A DECIMAL NUMBER AND PUSHED INTO THE STACK. SO AFTER THE READ Y= REG NUMBER, X=CONTENT OF THE REG

WFRM : WRITE FRAME (SENDING A FRAME)

Y IS TAKEN AS A DECIMAL NUMBER PRESENTING THE 8 DATA BITS. X IS TAKEN AS A NUMBER PRESENTING THE 3 CONTROL BITS. EXAMPLE: SENDING THE COMMAND FRAME "IFC" X = 4, Y = 144. "WFRM" WILL CHECK "ORAV" AND "FRNS" BEFORE SENDING THE FRAME. IF "FRNS" WAS SET, IT WILL GENERATE "FRNS ERR" MESSAGE. IF "ORAV" IS NOT SET WITHIN 10 SECONDS, IT WILL GENERATE "ORAV = 0" MESSAGE. "WFRM" WILL NOT WAIT FOR FRAME COMES BACK NOR WILL IT DO THE ERROR CHECKING AFTERWARD.

RFRM : READ FRAME

"RFRM" WILL READ RSGISTER 1&2 AND PUT THEM INTO STACK WHERE X = A DECIMAL NUMBER PRESENTING THE 3 CONTROL BITS AND Y = A DECIMAL NUMBER PRESENTING THE 8 DATA BITS OF THE FRAME. "RFRM" WILL NOT CHECK "ORAV" & "FRNS" PRIOR TO READ REGISTER 1 & 2. IT WOULD JUST READ IT.

THE FOLLOWING 5 FUNCTIONS ARE FOR TESTING THE LOWER 5 BITS OF REGISTER 1. IF IT IS EXECUTED FROM PROGRAM EXECUTION, IT WILL SKIP THE NEXT STEP IF THE ANSWER IS YES. IF IT IS EXECUTED FROM KEY BOARD, IT WILL ONLY DISPLAY "YES" OR "NO".

IFCR? : IFC RECEIVED ?

SRQR? : SERVICE REQUEST RECEIVED ?

FRAV? : FRAME AVAILABLE ?

FRNS? : FRAME RETURN NOT AS SEND ?

ORAV? : OUTPUT REGISTER AVAILABLE ?

THERE ARE TWO POTENTIAL PROBLEMS WHEN USE THIS DIAGNOSITC ROM WITH PIL BASIC MODULE. ONE IS THAT IF THE PIL CHIP IS NOT AUTO SOURCING "IDY", IT WILL BE TURN OFF WHEN THE 41C GOES TO LIGHT SLEEP. ANOTHER ONE IS THAT IF THE 41C IN AUTO I/O MODE, IT WILL CONSTANTLY SOURCE FRAMES TO LOOK FOR THE PRINTER. AND THIS WILL CHANGE THE REGISTER 0,1,2,3 OF THE PIL CHIP. IN ORDER TO PREVENT THE 41C GOING TO LIGHT SLEEP AND TO PREVENT THE BASIC MODULE MESSING AROUND WITH THOSE REGISTERS, WE PROVIDE THE FOLLOWING 3 FUNCTIONS :

SF33 : SET USER FLAG 33

WHEN USER FLAG 33 IS SET, NO PIL FUNCTION WILL BE EXECUTED BY THE BASIC MODULE. OF COURSE, IT WILL NOT LOOK FOR THE PRINTER EITHER. WHILE FLAG 33 IS SET, TRY TO EXECUTE ANY PIL BASIC FUNCTION WILL RESULT IN TRANSMIT ERROR. THE USER FLAG 31-35 CAN NOT BE SET OR CLEARED BY THE MAINFRAME "SF" AND "CF" FUNCTION. AND IT WILL NOT BE CLEARED ON TURN ON EITHER. THESE CAN BE ONLY CLEARED BY "MEMORY LOST" OR BY SOME SPECIAL FUNCTION LIKE THE NEXT FUNCTION "CF33".

CF33 : CLEAR USER FLAG 33

ON : STAY ON

ACTUALLY THIS IS A MAINFRAME FUNCTION TO PREVENT THE 41C GOING FROM LIGHT SLEEP TO DEEP SLEEP(OFF). THE "ON" FUNCTION ALSO SETS THE SYSTEM FLAG 44. WHEN FLAG 44 SET, THIS DIAGNOSTIC ROM WILL PREVENT THE 41C GOING INTO LIGHT SLEEP SO WE WE CAN MAKE THE PIL CHIP STAY ON ALL THE TIME. BUT THIS WILL MAKE THE 41C CONSTANTLY AWAKE AND DRAIN THE POWER IN A MUCH FASTER RATE TOO .

3.0 SENDING A FRAME BY ITS MNEMONIC

FOR CONVENIENCY, THERE IS A NUMBER OF FRAMES CAN BE SEND BY THEIR MNEMONIC. PRIOR TO SOURCE A FRAME, IF "ORAV" NOT GET SET WITHIN 10 SECONDS, IT WILL PRODUCE AN ERROR AND THE FRAME WILL NOT BE SENT. AFTER SENDING THE FRAME, IT WILL WAIT FOR THE FRAME COMES BACK AND DO THE ERROR CHECKING.

3.1 SENDING COMMAND FRAMES

THE FOLLOWING FUNCTIONS CAN BE USED TO SOURCE A COMMAND FRAME. SINCE ONLY CONTROLER CAN SOURCE COMMAND FRAME, SO "CA"(CONTROLER ACTIVE) IS AUTOMATICALLY SET TO 1 PRIOR SOURCING THE COMMAND FRAME. PRIOR TO SOURCE A COMMAND FRAME, IT ALWAYS CHECK "ORAV". IF "ORAV" NOT SET WITHIN 10 SECONDS, IT WILL GENERATE "ORAV = 0" ERROR. AFTER SENDING THE COMMAND FRAME, THE "RFC" WILL BE SOURCED AUTOMATICALLY BY THE CHIP. IF THE "RFC" DOES NOT RETURN WITHIN 10 SECONDS, IT WILL GENERATE "TIME OUT" ERROR.

AAU : (HEX 49A) AUTO ADDRESS UNCONFIGURE

CMD : SEND COMMAND
THE DIO8,DIO7,,,DIO1 OF THE THE COMMAND IS TAKEN FROM X-REG AS A DECIMAL NUMBER(0-255). FOR EXAMPLE: TO SEND COMMAND "DCL"(DEVICE CLEAR), SET X=20.

DDL : (HEX 4A0-4BF) SEND DEVICE DEPENDABLE LISTEN COMMAND
THE LOWER 5 BITS OF DDL COMMAND IS TAKEN FROM X-REG (0-31). IF X >= 32, WILL GENERATE "ADR ERR" MESSAGE. FOR EXAMPLE: TO SEND DDL 3, SET X = 3.

DDT : (HEX 4C0-4DF) SEND DEVICE DEPENDABLE TALK COMMAND
SAME AS DDL

GET : (HEX 408) GROUP EXECUTE TRIGGER

GTL : (HEX 401) GO TO LOCAL

IFC : (HEX 490) INTERFACE CLEAR
ONLY SYSTEM CONTROLER CAN SOURCE IFC, SO "SC" WILL BE SET TO 1 UPON EXECUTION OF THIS FUNCTION.

LAD : (HEX 420-43E) LISTEN ADDRESS
THE CONTENT OF X IS TAKEN AS THE ADDRESS NUMBER(<=30).

LPD : (HEX 49B) LOOP POWER DOWN

REN : (HEX 492) REMOTE ENABLE

SDC : (HEX 404) SELECTED DEVICE CLEAR

TAD : (HEX 440-45E) TALK ADDRESS
THE CONTENT OF X IS TAKEN AS THE ADDRESS NUMBER (<=30).

UNL : (HEX 437) UNLISTEN

UNT : (HEX 457) UNTALK

3.2 SENDING READY FRAMES

"ORAV" WILL BE CHECKED BEFORE SOURCING A READY FRAME AS IT DOES IN SOURCING A COMMAND FRAME. BUT THE TIME OUT AND ERROR CHECKING ARE NOT NECESSARILY PERFORMED, IT WILL DEPENDS ON WHAT FRAME IS BEING SOURCED."CA" IS ALWAYS SET TO PRIOR TO SEND A READY FRAME.

ASP : (HEX 580-59F) AUTO SIMPLE PRIMARY

THE CONTENT OF X IS TAKEN AS THE STARTING ADDRESS NUMBER(<=30). THE RETURN ADDRESS NUMBER IS PUSHED INTO STACK. "CA" IS SET TO 1 PRIOR TO SOURCE ASP.

THE FOLLOWING 4 FUNCTIONS "SDA", "SAC", "SID", "SSP" ARE CALL THE "SOT" FRAME (START OF TRANSMITION). IT ASSUME THE DEVICE IS A ADDRESSED TALKER AND WILL EAT THE "SOT" FRMAE AND START TO SOURCE DATA FRAME, IF THEY RECOGNIZED THESE FRAME. IF THE ADDRESSED TALKER DOES NOT RECOGNIZE THEM, IT SHOULD RETRANSMIT THE "SOT" FRAME.

SDA : (HEX 560) SEND DATA

THIS FUNCTION WILL ONLY SEND THE READY FRAME "SDA" BUT WILL NOT WAIT FOR ANY FRAME COMING BACK. "CA" IS SET TO 1 PRIOR TO SOURCE THE "SDA"

SAC : (HEX 563) SEND ACCESSORY BYTE

SET CA=1 AND SEND READY FRAME "SAC". THEN IT WILL READ THE INCOMMING DATA FRAMES UNTIL A NON-DATA FRAME COMES IN. ALL THE DATA FRAME WILL BE TREATED AS ONE BINARY NUMBER AND CONVERTED INTO A DECIMAL NUMBER AND PUSHED INTO STACK. IF THE LAST NON-DATA FRAME IS NOT THE "ETO", IT WILL GENERATE "EOT ERR". IF THE "SAC" BEEN RETRANSMITED BACK BY THE DEVICE, IT WILL GENERATE "NO RESPONSE" ERROR. IF NO FRAME COMES BACK WITHIN 10 SECONDS, WILL GENERATE "TIME OUT" THIS FUNCTION ASSUME THE DEVICE ALREADY BEEN ADDRESSED AS A TALKER. ERROR.

SID : (HEX 562) SEND DEVICE ID

SIMILER TO THE "SAC" FUNCTION EXCEPT THE DEVICE ID WILL BE STORED IN ALPHA REGISTER.

SSP : (HEX 561) SEND SERIEL POLL

SIMILER TO THE "SAC" FUNCTION EXCEPT THE READY FRAME SEND IS "SSP".

TCT : (HEX 564) TAKE CONTROL

THE WAY PASS CONTROL WORKS IS THAT THE ACTIVE CONTROLLER FIRST ADDRESS THE POTENTIAL CONTROLLER AS A TALKER, AND THEN SEND IT THIS "TCT" READY FRAME. IF THE POTENTIAL CONTROLLER WILL TAKE THE CONTROL, IT WILL EAT THE "TCT" FRAME AND START TO SOURCE ITS FIRST COMMAND FRAME. IF THE POTENTIAL CONTROLLER RETRANSMIT THE "TCT" READY FRAME, THAT MEANS IT WILL NOT TAKE THE CONTROL.

THE "TCT" FUNCTION ASSUMES THE POTENTIAL CONTROLLER ALRAEDY BEEN ADDRESSED AS A TALKER, SO IT START WITH SENDING THE "TCT" FRAME. THEN IT WILL WAIT FOR A INCOMMING FRAME. IF THE INCOMMING FRAME IS A COMMAND FRAME, IT WILL PUT THE COMMAND FRAME IN X & Y

REGISTER AS IT DOES IN "RFRM" FUNCTION. IF THE "TCI" COMES BACK OR NO FRAME COMES IN WITHIN 10 SECONDS, IT WILL GENERATE "NO RESPONSE" ERROR.

NRD : (HEX 542) NOT READY FOR DATA
THIS FUNCTION WILL SEND A SERIES OF FRAMES AND DO THE ERROR CHECKING. IT WILL FIRST READ THE DATA FRAME SUPPOSEDLY IN REGISTER 2 AND SAVE IT. THEN SOURCE THE NRD FRAME. WHEN THE NRD COMES BACK, IT WILL RETRANSMIT THE SAVED DATA FRAME, AND THEN EXPECT AN ETO/ETE COMES BACK. IF THE LAST FRAME RETURN IS NOT AN ETO, IT WILL GENERATE "EOT ERR".

3.3 SENDING IDY FRAME

1. IDY : SEND OUT IDY FRAME WITH DIO8-DIO1 EQUAL TO ZEROS.
"CA" IS SET TO 1 PRIOR SOURCING THE IDY FRAME. WHEN IT COMES BACK, DIO8-DIO1 ARE CONVERTED INTO A DECIMAL NUMBER AND PUSHED INTO STACK.

3.4 SEND/READ DATA FRAMES

OUTBIN : OUTPUT BINARY IN X
CONVERT THE NUMBER IN X TO BINARY AND SEND IT OUT AS DATA FRAMES. THE NUMBER OF BYTES TO SEND IS THE MINIMUM NUMBER OF BYTES WHICH CAN COVER THE X IN BINARY. IF THE CONTENT OF X IS A STRING, IT WILL ONLY SEND THE NON-NULL BYTES. IT ASSUMES THE DEVICE ALREADY BEEN ADDRESSED AS A LISTENER. "TA" IS SET TO 1 AND CHECK "ORAV" & "FRNS" PRIOR SOURCING THE DATA FRAMES. IF TRANSMIT ERROR IS DETECTED, IT WILL PRODUCE AN ERROR. AGAIN, THE TIME OUT IS SET TO 10 SECONDS. NO ETO/ETE WILL BE SENT AT THE END OF TRANSMITION.

OUTBINY : OUTPUT BINARY IN X AND COUNTED BY Y
SAME AS OUTBIN EXCEPT THE NUMBER OF BYTES TO SEND IS CONTROL BY THE NUMBER IN Y. BUT THE MAXIMUM NUMBER OF BYTES IS 7.

INBIN : INPUT BINARY TO X
SET LA=1 AND WILL SEND READY FRAME "SDA" PRIOR TO READ ANY FRAME. THIS FUNCTION WILL KEEP READING THE DATA FRMAE UNTIL A NON-DATA FRAME COMES IN. BUT IT WILL ONLY CONVERT THE LAST 7 BYTES OF DATA FRAME TO A DECIMAL NUMBER AND PUSH IT INTO STACK. IF THE LAST FRAME IS NOT AN "ETO", IT WILL PRODUCE "EOT ERR" MESSAGE. BUT THE X-REG WILL STILL BE CHANGED EVEN AN ERROR IS DETECTED. IF NO FRAME COMES IN WITHIN 10 SECONDS, WILL GENERATE "TIME OUT" ERROR.

4.0 HOW THE ERROR WILL AFFECT THE PROGRAM EXECUTION

THERE ARE FIVE KINDS OF ERROR : "TIME OUT", "FRNS ERR", "ETO ERR", "NO RESPONSE" AND "ORAV = 0" ARE HANDLED DIFFERENTLY FROM THE WAY THE 41C HANDLES ITS ERROR. IT CAN BE DESCRIBED BY THE FOLLOWING TABLE :

	FLAG 25 SET	FLAG 25 CLEAR
EXECUTE FROM KEY BOARD	X=NEGATIVE ERROR CODE : DISPLAY ERROR MESSAGE : WILL NOT CLEAR FLAG 25	X=NEGATIVE ERROR CODE : DISPLAY ERROR MESSAGE
EXECUTE FROM PROGRAM EXECUTION OR SST	X=NEGATIVE ERROR CODE : WILL NOT CLEAR FLAG 25 : PROGRAM CONTINUE	X=NEGATIVE ERROR CODE : DISPLAY ERROR MESSAGE : STOP PROGRAM EXECUTION

THE FIVE ERROR CODES ARE :

-1 : TIME OUT ERROR. TIME OUT IS ALWAYS SET TO 10 SECONDS

-2 : FRNS ERROR(FRAME RETURN NOT AS SEND)

-3 : EOT ERROR(END OF TRANSMITIUN ERROR)

-4 : NO RESPONSE ERROR. DEVICE DOES NOT RESPOND TO "SAC", "SID", "SSP" OR "TCI".

-5 : ORAV = 0 ERROR. PRIOR TO SOURCE A FRAME, "ORAV" DOES NOT GET WITHIN 10 SECONDS.

5.0 BUFFER OPERATION

A ONE DIMENSIONAL BYTE ARRAY CAN BE SET UP AS A BUFFER TO READ/WRITE A LARGER NUMBER OF FRAMES. A POINTER WILL BE USED AS THE INDEX OF THE BUFFER, WHICH CAN POINT TO ANY BYTE IN THE BUFFER.

SETBUFX : SET UP THE BUFFER BY X

THE CONTENT OF X IS TAKEN AS THE BUFFER SIZE IN BYTES. IF X = 0 THE EXISTING BUFFER, IF THERE IS ONE, WILL BE DISLOCATED. THE BUFFER IS CREATED AS AN I/O BUFFER IN 41C'S SPARE USER MEMORY. SO THE BUFFER SIZE IS LIMITED TO 1771 BYTES. BESIDES, IF THE BUFFER SIZE IS NOT EXACTLY AN MULTIPLE OF 7, IT WILL BE FILLED UP TO MULTIPLE OF 7 BYTES. IF THE 41C DOES NOT HAVE THE ENOUGH MEMORY TO ALOCATE THE BUFFER, IT WILL PACK THE MEMORY AND ASK THE USER TO TRY AGAIN.

PT= : SET THE POINTER TO A GIVEN BYTE NUMBER.

THE CONTENT OF X IS TAKEN AS THE POINTER VALUE. THE POINTER IS STARTING FROM ZERO. IF THE POINTER IS ASKED TO SET BEYOND THE END OF BUFFER, IT WILL PRODUCE AN ERROR.

PT=? : ASK THE PRESENT VALUE OF THE POINTER

THE PRESENT VALUE OF THE POINTER WILL BE PUSHED INTO STACK.

AIPT : AUTO ADVANCE POINTER AFTER EACH BUFFER OPERATION

THE BUFFER POINTER CAN BE AUTOMATICALLY ADVANCED AFTER EACH BUFFER OPERATION BY SELECTING AUTO ADVANCE MODE. THIS MODE WILL STAY IN EFFECT UNTIL THE OTHER FUNCTION "MIPT" IS EXECUTED.

MIPT : MANUAL ADVANCE POINTER

WHILE IN MANUAL ADVANCE MODE, POINTER WILL STAY WHERE IT WAS AFTER EACH BUFFER OPERATION. THE POINTER WILL ONLY BE MOVED BY THE FUNCTION "PT=". WHEN THE BUFFER FIRST SET UP, IT IS ASSUMED IN MANUAL ADVANVE MODE.

X-BUF : STORE X TO BUFFER

THE CONTENT OF X WILL BE CONVRETED INTO BINARY AND STORED INTO BUFFER STARTING FROM WHERE THE POINTER IS. THE NUMBER OF BYTES IS DETERMINED BY THE VALUE OF X. IF X HAS A STRING, IT WILL SUPRESS NULL BYTES AND NO CONVERSION WILL BE DONE.

BUF-XA : CONVERT ASCII CODED DECIMAL NUMBER FROM BUFFER TO X

THE NUMBER IN X-REG IS TAKEN AS THE NUMBER OF BYTS TO CONVERT. STARTING FROM WHERE THE POINTER IS, CONVERT X NUMBER OF BYTES, DECIMAL NUMBER IN ITS ASCII FORM, INTO A DECIMAL NUMBER AND PUSH IT INTO STACK. IT WILL BE TERMINATED BY "CR" &"LF".

BUF-XB : CONVERT BINARY TO DECIMAL FROM BUFFER

SIMILAR TO BUF-XA, EXCEPT THE BYTES PICK UP FROM BUFFER ARE TREATED AS A BINARY NUMBER.

A-BUF : STORE ALPHA REGISTER TO BUFFER

STORE THE CONTENT OF ALPHA REGISTER TO BUFFER STARTING FROM WHERE THE POINTER IS. IF BUFFER OVERFLOWED, IT WILL PRODUCE AN ERROR.

BUF-AX : RESTORE ALPAH REGISTER FROM BUFFER BY X

RESTORE X NUMBER OF BYTES FROM BUFFER TO ALPHA REGISTERE STARTING FROM THE POINTER. THE CONTENT OF X IS TAKEN AS BYTE COUNT. THIS

FUNCTION WILL ALSO ■■ PRE-TERMINATED BY "CR","LF" TOO. IT WILL NOT CLEAR ALPHA REGISTER BEFORE EXECUTION.

INBUFX : INPUT DATA FRAMES INTO BUFFER BY X
X IS TAKEN AS BYTE COUNT. SET LA=1 AND WILL SEND "SDA" PRIOR TO READ. READS X NUMBER OF DATA FRAMES AND STORE THEM TO BUFFER STARTING FROM THE CURRENT POINTER. THE EXECUTION WILL BE TERMINATED AS FOLLOWAS:

1). BY A NON-DATA FRAME, BUT IF IT IS NOT AN "ETO" WILL GENERATE AN ERROR.

2). READ IN X NUMBER OF BYTES. AND IT WILL SEND THE "NRD" AFTER LAST BYTE.

3). REACHED END OF BUFFER. AN ERROR WILL BE GENERATED. TIME OUT IS SET TO 60 SECONDS FOR THIS FUNCTION, BUT ANY KEY DOWN WILL SHORT CIRCUIT THE TIME OUT AND GENERATE "TIME OUT" ERROR.

OUTBUFX : OUTPUT DATA FRAME FROM BUFFER BY X
X IS TAKEN AS BYTE COUNT. SET TA=1 PRIOR TO EXECUTION. THEN SENDS X NUMBER OF DATA FRAMES FROM BUFFER STARTING FROM THE CURRENT POINTER POSITION. THIS FUNCTION WILL HANDLE "NRD" PROPERLY IN MIDDLE OF THE TRANSMITION AND WILL SEND AN ETO/ETE AT THE END OF TRANSMITION.

RG-BUFX : COPY REGISTERS TO BUFFER BY ■
THE CONTENT OF X IS TAKEN AS THE REGISTER INDEX IN THE bbb.eee FORM, WHERE bbb IS THE BEGINNING REGISTER NUMBER AND eee IS THE ENDING REGISTER NUMBER. THE COPY WILL START FROM WHERE THE POINTER IS. IF THE BUFFER IS OVERFLOWED, IT WILL PRODUCE AN ERROR. IF ALL OR PARTIAL OF THE DATA REGISTER IS NOT EXIST, IT WILL PRODUCE AN ERROR. THE REGISTER WILL BE COPIED EXACTLY THE WAY IT IS, NO CONVERSION WILL BE MADE.

BUF-RGX : COPY BUFFER TO REGISTERS BY X
SAME AS "RG-BUF" EXCEPT THE DIRECTION IS REVERSED.

X=BUF? : COMPARE X WITH BUFFER
IF X HAS A NUMBER, IT WILL CONVERT THE ABS(X) INTO ■ BINARY NUMBER AND COMPARE THE SAME NUMBER OF BYTES WITH THE BUFFER STARTING FROM THE CURRENT POINTER. IF ■ HAS A STRING, IT WILL COMPARE ALL THE NON-NULL BYTES WITH THE SAME NUMBER OF BYTES IN BUFFER. LIKE THE OTHER COMPARE FUNCTION, IT WILL DISPLAY "YES" OR "NO", OR SKIP THE NEXT STEP.

A=BUFX? : COMPARE ALPHA REGISTER TO BUFFER BY X
COMPARE ■ BYTES OF ALPHA REGISTER WITH BUFFER STARTING FROM POINTER. IF ■ IS LARGER THAN THE LENGTH OF ALPHA REGISTER, IT WILL ONLY COMPARE TO THE END OF ALPHA REGISTER.

A=BUF? : COMPARE ALPHA REGISTER TO BUFFER
SAME AS "A=BUFX?" EXCEPT IT ALWAYS COMPARE FROM THE BEGINNING TO THE END OF THE ALPHA REGISTER WITH THE SAME NUMBER OF BYTES IN BUFFER.

RG=BUF? : COMPARE REGISTERS TO BUFFER
THE CONTENT OF X IS TAKEN AS REGISTER INDEX IN THE bbb.eee FORM.

THE BLOCK OF REGISTERS WILL BE COMPARED WITH THE SAME NUMBER OF BYTES IN THE BUFFER STARTING FROM THE CURRENT POINTER POSITION. EVERY REGISTER IS DIRECTLY COMPARED WITH 7 BYTES IN THE BUFFER, NO CONVERSION WILL BE DONE.

6.0 SCOPE MODE

6.1 SCOPE MODE FUNCTIONS

SCOPE : PUT THE 41C INTO SCOPE MODE
IN SCOPE MODE, THE 41C WILL NO LONGER SOURCE FRAMES, BUT IT CAN DISPLAY OR BUFFER ANY FRAME PASSING BY. WHEN ENTERING THE SCOPE MODE, "TA" & "LA" ARE SET TO 1. ALL THE FRAME WILL BE SHOWN IN ITS MENUMONIC. THE DELAY OF EACH FRAME CAN BE CONTROLED BY THE DIGIT KEYS 0,1,2 AND 3. BUT THE IDY FRAME WILL ALWAYS USE ZERO DELAY. WHILE ANY KEY IS DOWN, THE TRANSMITION WILL BE HALT UNTIL THE KEY IS UP AGAIN.

MONITOR : PUT 41C INTO MONITOR MODE
THE MONITOR MODE IS LITTLE DIFFERENT FROM THE SCOPE MODE. IN MONITOR MODE THE 41C CAN STILL BE USED AS A REGULAR 41C. ALL THE MONITOR MODE DOES IS SET TA & LA TO 1 AND NEVER LET 41C GO INTO LIGHT SLEEP, AND IT CONSTANTLY CHECK IF THERE IS ANY FRAME COMES IN. WHEN THERE IS A FRAME COMES IN, IT WILL READ IT AND PUT IT TO 41C X & Y REGISTER LIKE THE "RFRM" FUNCTION DOES AND IT WILL SHOW THE FRAME IN THE DISPLAY. BUT THE FRAME WILL NOT BE RETRANSMITED AUTOMATICALLY. THE USER WILL HAVE TO EXECUTE THE "WFRM" FUNCTION TO RETRANSMIT THE FRAME. TO GET OUT OF THE MONITOR MODE, THE USER CAN EITHER CLEAR TA OR LA, OR WRITE 00 TO REGISTER 5 OF THE PIL CHIP.

PRFRMS : PRINT FRAMES IN BUFFER
PRINT THE FRAMES IN BUFFER STARTING FROM WHERE THE POINTER IS. EVERY FRAME IS USED TWO BYTES TO STORE IN BUFFER, SO THE POINTER NEED TO BE SET TO THE FIRST BYTE OF A FRAME.
THE PRINTING WILL BE STOPPED BY EITHER REACHING THE END OF BUFFER OF BY HITTING THE "R/S" KEY. IF ANY OTHER KEY IS PUSHED DURING THE PRINTING WILL SLOW DOWN THE PRINTING TO ABOUT 2 SECONDS PER FRAME. SO IF THERE IS NO PRINTER PLUG-IN, THE USER CAN USE THIS FEATURE TO SLOW DOWN THE DISPLAY TO A READABLE RATE.

PRBYTES : PRINT BYTES IN BUFFER
SIMILAR TO THE "PRFRMS" EXCEPT IT PRINTS EVERY SINGLE BYTE IN THE BUFFER STARTING FROM THE CURRENT POINTER.

6.2 SCOPE MODE KEY BOARD

WHEN GOES INTO SCOPE MODE, THE ROM WILL TAKE THE FULL CONTROL OF THE KEY BOARD. THE 41C WILL NEVER GO TO LIGHT SLEEP NOR THE CONTROL WILL PASS BACK TO THE 41C UNTIL EXIT FROM SCOPE MODE. ONLY THE FOLLOW KEYS WILL BE ENABLED IN SCOPE MODE:

1. "STO" KEY
THE STO KEY IS USED AS A TOGGLE KEY TO GO IN OR OUT OF BUFFER MODE. WHEN IT IS IN BUFFER MODE, ALL THE FRAMES PASSED BY WILL BE STORED INTO BUFFER STARTING FROM THE POINTER. TWO BYTES ARE REQUIRED FOR EVERY FRAME. WHILE IN BUFFER MODE, THE FRAME WILL STILL BE SHOWN AND THE DELAY SET PREVIOUSLY WILL STILL VALID. WHEN FILL UP TO THE LAST BYTE IN BUFFER, A MESSAGE "BUFFER FULL" WILL BE SHOWN IN DISPLAY, THE TRANSMITION WILL BE HALT AT THIS POINT. TWO THINGS THE USER CAN DO TO RESUME THE TRANSMITION:

1). HIT THE "STO" KET TO GET OUT OF BUFFER MODE.
2). HIT "BACK ARROW" KEY TO RESET THE POINTER TO ZERO.

2. DIGIT KEY "0"
SET THE FRAME DELAY TO ZERO. IN ORDER TO SHOW THE TRANSMITION THE FRAME WILL STILL BE ROLLING INTO THE DISPLAY, BUT IT WOULD JUST TOO FAST TO TELL WHAT IT IS.

3. DIGIT KEY "1"
SET FRAME DELAY TO 1 SECOND

4. DIGIT KEY "2"
SET FRAME DELAY TO 2 SECONDS

5. DIGIT KEY "3"
SET FRAME DELAY TO 3 SECONDS

6. "BACK ARROW" KEY
RESET THE BUFFER POINTER TO ZERO

7. "SST" KEY
MOVE THE POINTER TO NEXT FRAME AND DISPLAY IT. A MESSAGE "END OF BUFFER" WILL BE SHOWN WHEN REACH THE LAST FRAME IN THE BUFFER. FOR EVERY "SST", THE POINTER IS ADVANCED TWO BYTES.

8. "BST" KEY (SHIFT,SST)
MOVE THE POINTER BACK ONE FRAME AND DISPLAY IT

9. "R/S" KEY
EXIT FROM SCOPE MODE. WHEN EXIT FROM SCOPE, TA & LA ARE SET TO ZERO. IF THERE IS NO KEY DOWN AND NO FRAME TRANSMITING FOR 10 MINUTES, THE SCOPE MODE IS AUTOMATICALLY TERMINATED. WHENEVER EXIT THE SCOPE MODE, THE BUFFER POINTER WILL PROPERLY UPDATED IF IT IS IN POINTER AUTO ADVANCE MODE.

10. "OFF" KEY
CALCULATOR OFF

7.0 RECEIVING SERVICE REQUEST

THERE IS A WAY PROVIDED BY THIS ROM TO HANDLE SERVICE REQUEST AS AN INTERRUPT. THE 41C IS CAPABLE OF AUTOMATICALLY SOURCING IDY FRAME WHILE IT IS IN LIGHT SLEEP. IF THE 41C IS WAKEN UP FROM LIGHT SLEEP BY A SERVICE REQUEST, THE PRINTER CODE AT LOCATION @60000 WILL HAVE THE FIRST CHANCE TO SEE IF IS THE PRINTER REQUESTING SERVICE. IF IT IS, IT WILL SERVICE THE PRINTER. IF IT IS NOT THE PRINTER REQUESTING SERVICE, THEN THE FOLLOWING PLUG-IN ROM WILL HAVE THE CHANCE TO HAVE THE CONTROL. WHENEVER THIS ROM HAVE A CHANCE TO SERVICE A SERVICE REQUSET, IT WILL FIRST LOOK AT THE INTERRUPT ENABLE FLAG WHICH IS THE USER FLAG 18. IF THE USER FLAG 18 IS SET, THEN IT WILL TRY TO EXECUTE THE SERVICE ROUTINE. IF IT FINDS THE SERVICE ROUTINE, IT WILL JUMP TO START EXECUTING THE SERVICE ROUTINE. THE FIRST LINE OF THE SERVICE ROUTINE HAS TO AN ALPHA LABEL "SRQR". THAT IS THE UNIQUE LABEL IT WILL BE SEARCHING FOR. SO IF EITHER THE USER FLAG 18 IS NOT SET OR THE LABEL "SRQR" NOT FOUND, IT WILL IGNORE THE SERVICE REQUSET.

7.0 UTILITY FUNCTIONS

IOFSX? : TEST 41C CPU IO/FLAG BY X
THERE ARE 14 (0-13) HADRWARE IO/FLAG IN 41C CPU WHICH CAN BE TEST FOR SET OR CLEAR BY THIS FUNCTION. IF EXECUTE FROM KEY BOARD, IT WILL DISPLAY "YES" OR "NO". IF EXECUTE FROM PROGRAM, IT WILL SKIP NEXT THE NEXT STEP IF THE FLAG IS SET. THE CONTENT OF X IS USED TO INDICATE WHICH FLAG TO TEST.

ROMCHKX : VERIFY PLUG-IN ROM CHECKSUM BY X
X IS TAKEN AS THE ROM ID OF A PLUG-IN ROM. FOR EXAMPLE: THE ROM ID OF THE PIL MODULE IS 28 & 29. WHILE COMPUTING THE ROM CHECKSUM, THE DISPLAY WILL SHOW "DD CC-NN TST" WHERE DD IS THE ROM ID, CC-NN IS THE ROM LABEL. WHEN THE CHECKSUM COMPUTATION IS DONE, THE DISPLAY WILL CHANGE TO "DD CC-NN OK" OR "DD CC-NN BAD". IF THE FUNCTION WAS EXECUTED FROM PROGRAM EXECUTION, IT WILL SKIP THE NEXT STEP IF THE CHECKSUM IS GOOD.

X<>FLAG : TAKE THE INT(X) AND EXCHANGE IT WITH USER FLAG 0-7
CONVERT INTEGER PART OF X (0-255) AND SAVE IT IN USER FLAG 0-255, THEN CONVERT THE USER FLAG 0-7 AS A DECIMAL NUMBER AND PUSH IT INTO STACK.